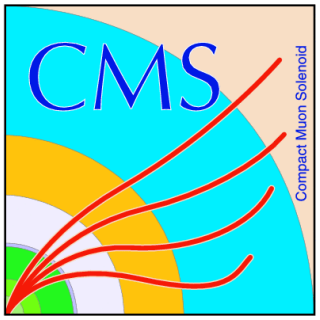
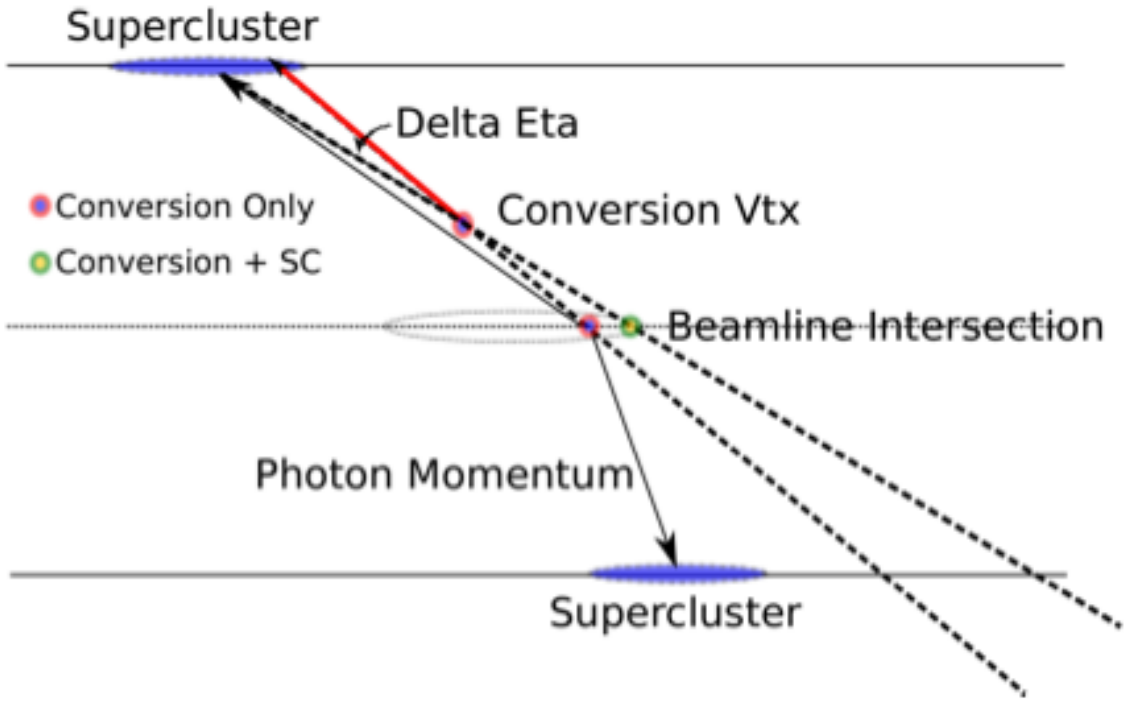
Updates on H→γγ Vertex ID Optimization

Julia Gonski (Dr. Pasquale Musella)
CMS Hgg Sub-group
19 July 2013



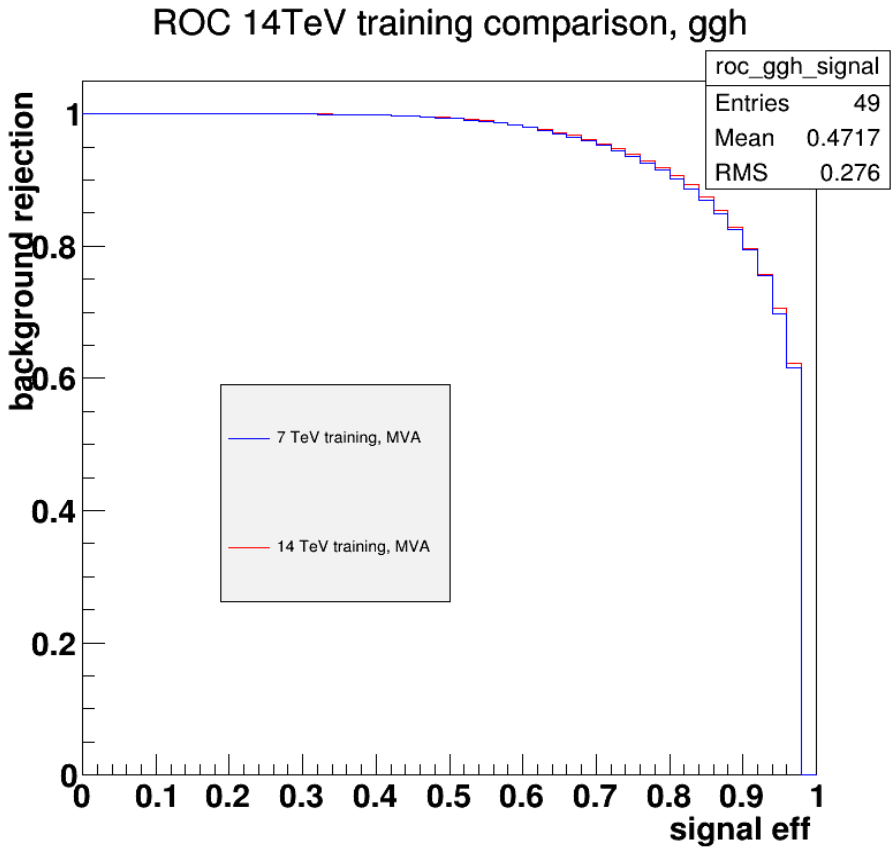
From last time…

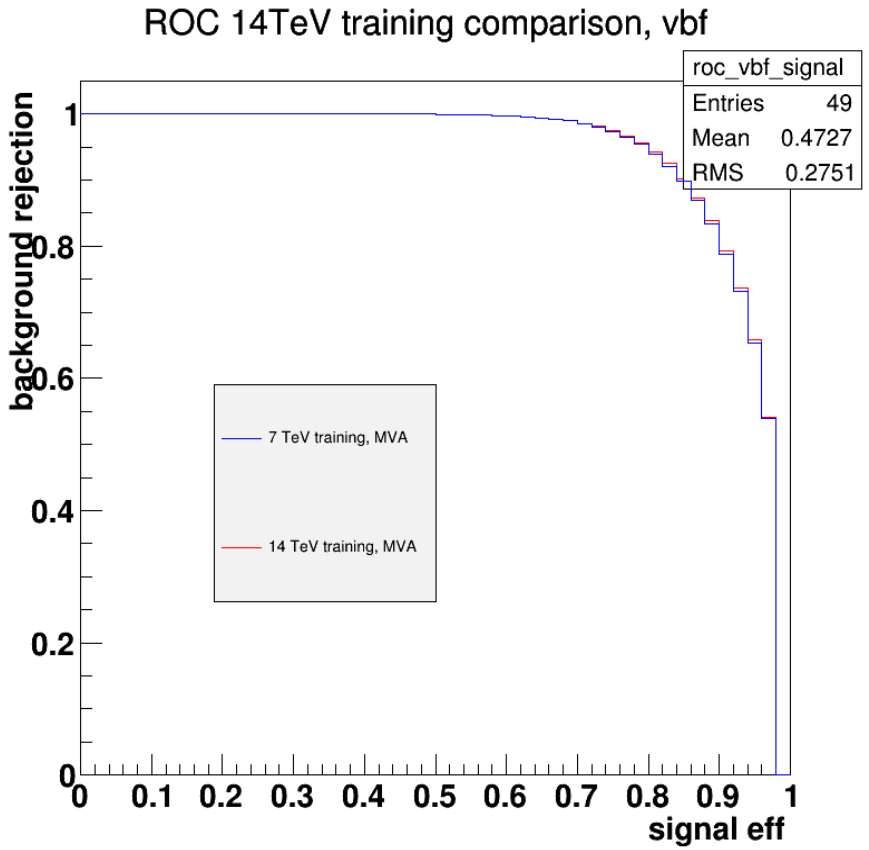
- Goal: determine Higgs to gamma gamma vertex position within **1cm** for mass resolution improvement
- Retrain Boosted Decision Tree (BDT) for new 14 TeV pile up conditions
- Develop time of flight (tof) discriminant to improve vertex ID efficiency



BDT 14 TeV re-training

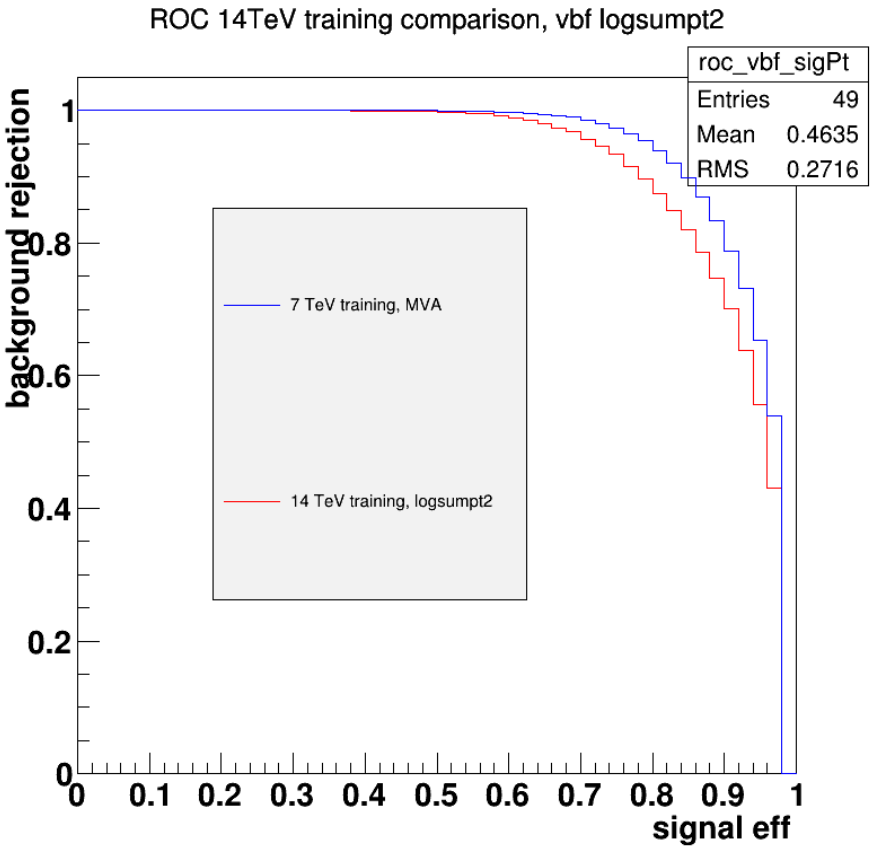
- Pile up simulated as Gaussian centered around mean of 50 PU events
- ROC comparison: background rejection vs. signal efficiency

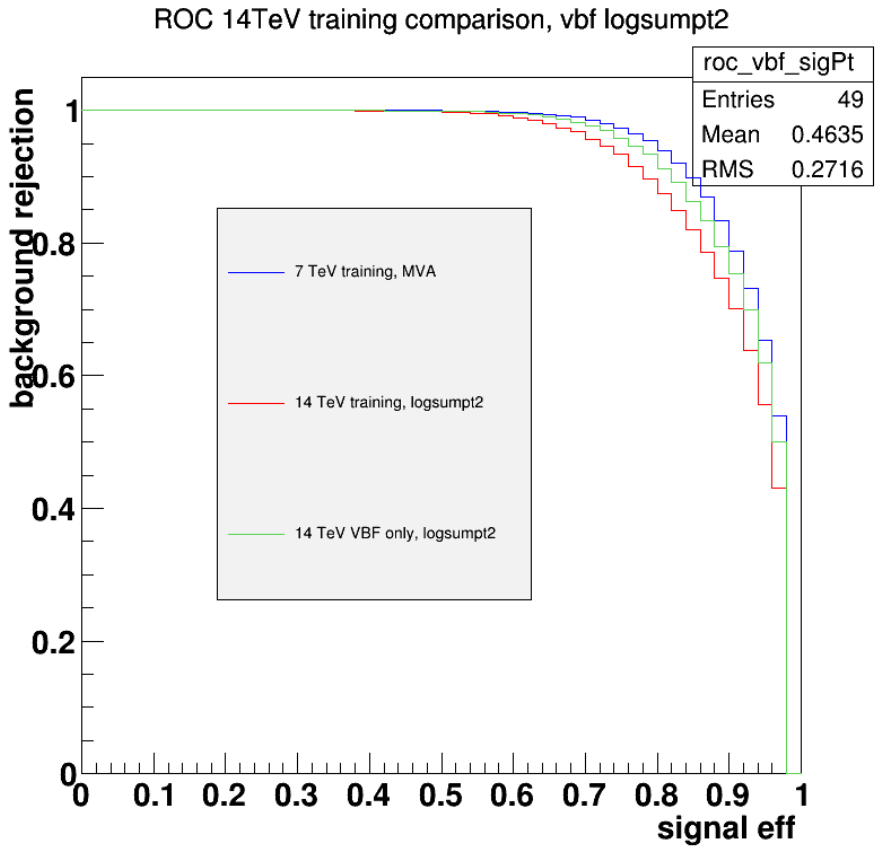




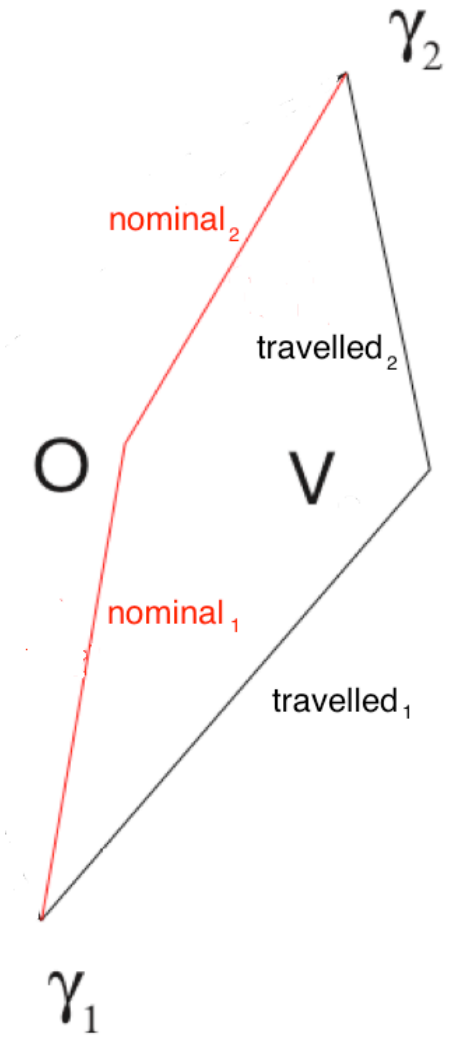
BDT 14 TeV re-training: VBF focus

- VBF: special case for one or more jets missing detector, false p_T offset
- Train VBF events using $\sum_i |\vec{p}_T^{\,i}|^2$ as a discriminant





TOF Discriminant Computation



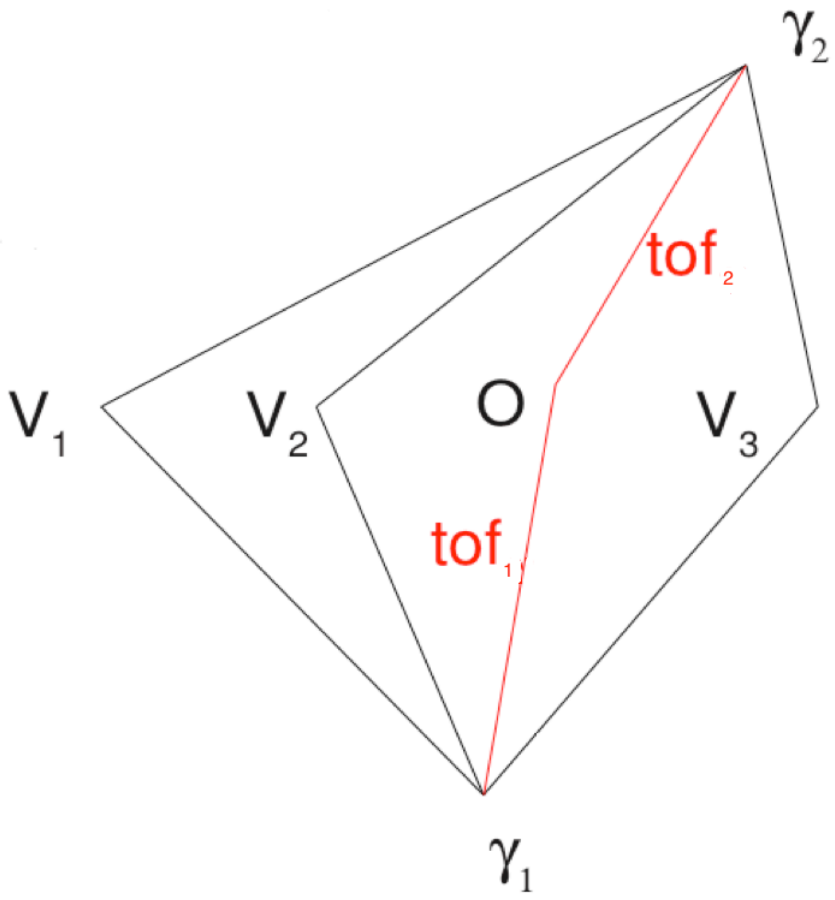
- Detector assumes that all vertices are in the center of the detector (O)
- Calibration required to account for z displacement for each vertex

$$TOF = \frac{travelled - nominal}{c}$$

- For comparison to other vertices, compute difference between TOF for each photon

$$\Delta TOF = TOF(\gamma 1) - TOF(\gamma 2)$$

TOF Discriminant Calculation, cont.



- Correct vertex determined in test tree
- Training: iterate through candidate vertices and compute ΔTOF for comparison
- **Match: ΔΔTOF = 0**

$$\Delta\Delta TOF = \Delta TOF(O) - \Delta TOF(V)$$

Next steps

- Finish incorporation of TOF discriminant: better choice of metric?
- Check performance ROC curves
- Optimize based on Δeta between two photons: include as discriminant in BDT training
- Train for increased pile up (~100 events?)

Backup

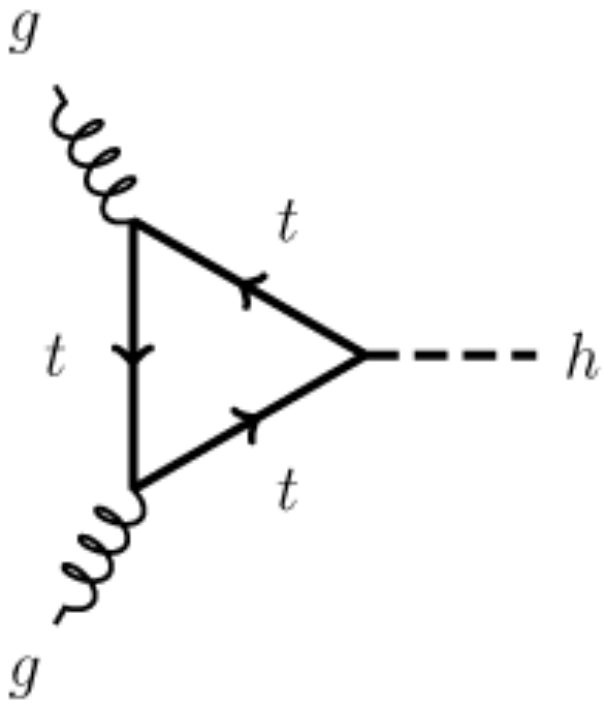
Analysis Code: Off-center calibration

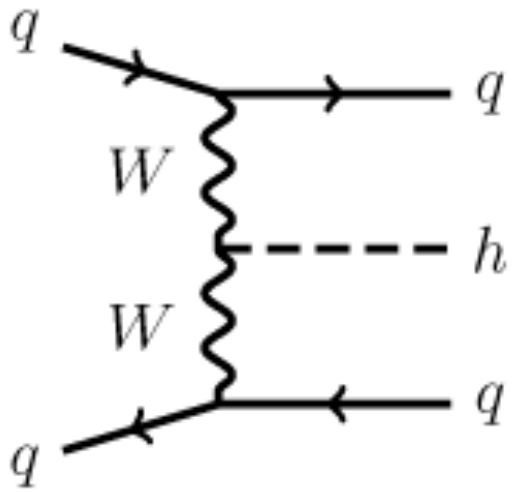
```
float VertexOptimizationAnalysis::getExtraTravelTime(TVector3 &posSC, TVector3 &posVertex){
  float travelled = sqrt( pow(posSC.X()-posVertex.X(), 2) +
            pow(posSC.Y()-posVertex.Y(), 2) +
            pow(posSC.Z()-posVertex.Z(), 2)  );      //from true vertex
  float nominal   = sqrt( pow(posSC.X(), 2) +
            pow(posSC.Y(), 2) +
            pow(posSC.Z(), 2)  );      //from origin of detector

  return (travelled-nominal)/100./speedOfLight*1.e9;
              //returns calibration time in nanoseconds
}

float VertexOptimizationAnalysis::getDeltaTof(TVector3 &posLead, TVector3 &posSubLead,
TVector3 &posVertex){

return getExtraTravelTime(posLead,posVertex) - getExtraTravelTime(posSubLead,posVertex);
            //computes difference in travel time between two photons
}
```

Analysis code: vertex loop

```
TVector3 caloPosLead    = ( * (TVector3*) l.pho_calopos-
>At(  l.dipho_leadind[diphoton_id] ) ) ;                    //photon 1

TVector3 caloPosSubLead = ( * (TVector3*)
l.pho_calopos->At(  l.dipho_subleadind[diphoton_id] ) ) ; //photon2

TVector3 closestVertex  = ( * (TVector3*)l.vtx_std_xyz->At(closest_id) );     //correct vertex

deltaTof = getDeltaTof(caloPosLead, caloPosSubLead, closestVertex); //between 2 photons of
correct vertex

deltaTof += getTimeResol(timeResVal_); //add smearing factor for time resolution


for(int vi=0; vi<l.vtx_std_n; ++vi) {
      … //other analysis code
    TVector3 currentVertex  = ( * (TVector3*)l.vtx_std_xyz->At(vi) );
    tofCorrTdiff_ = deltaTof -  getDeltaTof(caloPosLead, caloPosSubLead, currentVertex);
            //compute difference in diphoton tof between correct and current
    l.FillTreeContainer("vtxOpt");
}
```

Higgs production





Variable Definitions

- *sumpt2*: $\sum_i |\vec{p}^{\,i}_T|^2$.
- *ptbal*: $-\sum_i(\vec{p}^{\,i}_T \cdot \frac{\vec{p}^{\,\gamma\gamma}_T}{|\vec{p}^{\,\gamma\gamma}_T|})$.
- *ptasym*: $(ptvtx - p^{\gamma\gamma}_T) \,/\, (ptvtx + p^{\gamma\gamma}_T)$.